Implementasi Algoritma Jaro-Winkler Distance untuk Membandingkan Kesamaan Dokumen Berbahasa Indonesia

[1]Anna Kurniawati
[2]Sulistyo Puspitodjati
[3]Sazali Rahman

[1]Universitas Gunadarma(ana@staff.gunadarma.ac.id)

[2]Universitas Gunadarma(sulistyo@staff.gunadarma.ac.id)

[3]Universitas Gunadarma(sazali@yahoo.com)

Ringkasan

Seiring dengan berkembangnya dunia pendidikan, semakin banyak pula pembuatan karya penelitian atau penulisan ilmiah mahasiswa. Dengan semakin banyaknya penulisan ilmiah yang dilakukan oleh mahasiswa, tidak menutup kemungkinan jika terdapat penulisan yang sama. Untuk dapat mengukur tingkat kesamaan dokumen dengan cepat, maka diperlukan alat bantu untuk dapat menghitung tingkat kesamaan antar dokumen. Pada makalah ini akan dipaparkan pembuatan aplikasi menghitung tingkat kesamaan dokumen berbasis web. Metode yang digunakan untuk menghitung tingkat kesamaan dokumen dengan menggunakan Algoritma jaro-winkler distance. Pembuatan aplikasi ini menggunakan bahasa pemrograman PHP dan perangkat lunak basis data yang digunakan adalah MySql. Pengujian terhadap aplikasi ini dengan menggunakan data abstraksi penulisan atau penelitian ilmiah jurusan Sistem Informasi Universitas Gunadarma sebanyak 50 abstraksi.
Kata kunci : dokumen, kesamaan, algoritma jaro-winkler distance.

1 Pendahuluan

Dengan semakin berkembangnya teknologi informasi, sehingga membuat pembuatan karya tulis semakin mudah dan cepat. Hal tersebut dikarenakan informasi kini tersedia secara melimpah. Akan tetapi dikarenakan kemudahan dalam memperoleh informasi tersebut, pada pembuatan karya tulis sering ditemukan kesamaan dengan karya tulis orang lain sehingga kemudian menimbulkan isu plagiarisme.

Aksi plagiat dalam karya tulis sangatlah mungkin terjadi, untuk itu perlu dilakukan upaya-upaya sebagai pencegahan maupun pendeteksian, sehingga dalam tulisan ini akan dibahas mengenai pendeteksian plagiarisme dari sebuah dokumen dengan membandingkannya dengan sebuah dokumen lainnya. Karya tulis selanjutnya akan disebut sebagai dokumen teks.

Untuk mengetahui seberapa besar kesamaan suatu dokumen teks dengan dokumen teks lainnya dapat dengan menggunakan pendekatan string metric yaitu melakukan perbandingan string dengan memasukkannya ke dalam fungsi matematis tertentu. Beberapa algoritma yang berdasarkan kepada string metric diantaranya adalah Levenshtein distance, TF/IDF, Needleman-Wunsch distance, Jaro-Winkler distance, dan sebagainya. Dari algoritma yang telah disebutkan di atas Jaro-Winkler distance memiliki ketepatan yang baik di dalam pencocokan string yang relatif pendek. Metode ini dipilih dikarenakan setelah dilakukannya proses tokenizing algoritma ini dapat secara akurat memeriksa salinan antar dokumen. Diharapkan dengan adanya sebuah aplikasi yang mampu mendeteksi kesamaan dokumen maka pada tulisan ilmiah tidak ditemukan lagi kesamaan yang begitu tinggi dengan dokumen lainnya.

2 Teori

2.1 Deteksi Plagiarisme

Sebagian besar kasus plagiarisme ditemukan di bidang akademisi, biasanya berupa essai, jurnal, laporan penelitian, dsb. Namun kini, plagiarisme dapat ditemukan hampir di semua bidang, termasuk karya ilmiah, seni desain, dan source code. Oleh sebab itu, banyak peneltian yang memfokuskan diri ke dalam pendeteksian plagiarisme, seperti deteksi pada dokumen teks ataupun source code. De-

teksi plagiarisme biasanya dilakukan dengan membandingkan sebuah dokumen dengan dokumen lainnya. Tingkat kesamaan dari dokumen tersebutlah yang akan menjadi dasar untuk pendeteksian plagiarisme. Di dalam tulisan ini dipaparkan sebuah aplikasi untuk mendeteksi kesamaan pada dokumen teks dengan menggunakan algoritma Jaro-Winkler.

2.2 String Metric

String metric atau similarity metric adalah kelas matriks berbasis tekstual yang dapat menghasilkan nilai kesamaaan atau ketidaksamaan dari dua teks string untuk proses perbandingan dan penyamaan. String metric biasanya digunakan dalam deteksi kecurangan, analisa fingerprint, deteksi plagiarisme, ontology merging, analisis DNA , analisis RNA, analisis image, database deduplication, data mining, Web interfaces, dan sebagainya. Beberapa algoritma yang berdasarkan kepada string metric diantaranya adalah Levenshtein distance, TF/IDF, Needleman-Wunsch distance, Jaro-Winkler distance, dan sebagainya. Dari algoritma yang telah disebutkan di atas Jaro-Winkler distance memiliki ketepatan yang baik di dalam pencocokan string yang relafif pendek. Untuk dapat mendukung kinerja dari algoritma Jaro-Winkler distance maka dilakukan proses tokenizing terlebih dahulu terhadap dokumen-dokumen yang akan digunakan.

2.3 Algoritma Jaro-Winkler

Jaro-Winkler distance adalah merupakan varian dari Jaro distance metrik yaitu sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi Jaro-Winkler distance untuk dua string, semakin mirip dengan string tersebut. Jaro-Winkler distance terbaik dan cocok untuk digunakan dalam perbandingan string singkat seperti nama orang. Skor normalnya seperti 0 menandakan tidak ada kesamaan, dan 1 adalah sama persis.

Algoritma Jaro-Winkler distance memiliki kompleksitas waktu quadratic runtime complexity yang sangat efektif pada string pendek dan dapat bekerja lebih cepat dari algoritma edit distance. Dasar dari algoritma ini memiliki tiga bagian:

1. Menghitung panjang string,
2. Menemukan jumlah karakter yang sama di dalam dua string, dan
3. Menemukan jumlah transposisi.

Pada algoritma Jaro digunakan rumus untuk menghitung jarak (d_j) antara dua string yaitu s_1 dan s_2 adalah

$$d_j = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \quad (1)$$

dimana :

m	=	jumlah karakter yang sama persis
$\|s_1\|$	=	panjang string 1
$\|s_2\|$	=	panjang String 2
t	=	jumlah transposisi

Jarak teoritis dua buah karakter yang disamakan dapat dibenarkan jika tidak melebihi:

$$\left(\frac{max\left(|s_1|, |s_2|\right)}{s} \right) < -1 \quad (2)$$

Akan tetapi bila mengacu kepada nilai yang akan dihasilkan oleh algoritma Jaro-Winkler maka nilai jarak maksimalnya adalah 1 yang menandakan kesamaan string yang dibandingkan mencapai seratus persen atau sama persis. Biasanya s1 digunakan sebagai acuan untuk urutan di dalam mencari transposisi. Yang dimaksud transposisi di sini adalah karakter yang sama dari string yang dibandingkan akan tetapi tertukar urutannya.. Sebagai contoh, dalam membandingkan kata CRATE dengan TRACE, bila dilihat seksama maka dapat dikatakan semua karakter yang ada di s1 ada dan sama dengan karakter yang ada di s2 , tetapi dengan urutan yang berbeda. Dengan mengganti C dan T, dapat dilihat perubahan kata CRATE menjadi TRACE. Pertukaran dua elemen string inilah adalah contoh nyata dari transposisi yang dijelaskan. Dalam pencocokkan DwAyNE dan DuANE memiliki urutan yang sama D-A-N-E, jadi tidak ada transposisi.

aro-Winkler distance menggunakan prefix scale (p) yang memberikan tingkat penilaian yang lebih, dan prefix length (l) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari string yang dibandingkan sampai ditemukannya ketidaksamaan. Bila string s_1 dan s_2 yang diperbandingkan, maka Jaro-Winkler distancenya (d_w) adalah:

$$d_w = d_j + (lp\,(1 - d_j)) \quad (3)$$

dimana :

d_j	=	Jaro distance untuk strings s_1 dan s_2
l	=	panjang prefiks umum di awal string nilai maksimalnya 4 karakter(panjang karakter yg sama sebelum ditemukan ketidaksamaan max 4)
p	=	konstanta scaling factor. Nilai standar untuk konstanta ini menurut Winkler adalah p = 0,1.

Berikut ini adalah contoh pada perhitungan Jaro-Winkler distance. Jika string s_1 MARTHA dan s_2 MARHTA maka:

m = 6
s_1 = 6
s_2 = 6

Karakter yang tertukar hanyalah T dan H. Maka $t = 1$.

Maka nilai Jaro distance adalah:

$$d_j = \frac{1}{3} \times \left(\frac{6}{6} + \frac{6}{6} + \frac{6-1}{6} \right) = 0.944$$

Kemudian bila diperhatikan susunan s_1 dan s_2 dapat diketahui nilai $l = 3$, dan dengan nilai konstan $p = 0.1$. Maka nilai Jaro-Winkler distance adalah:

$$d_w = 0.944 + (3 \times 0.1\,(1 - 0.944)) = 0.961$$

Jika string s_1 DWAYNE dan s_2 DUANE maka:

m = 4
s_1 = 6
s_2 = 5

t = 0, hal ini dikarenakan tidak ada karakter yang sama tapi tertukar urutannya. Karakter seperti D, A, N, E dianggap dalam urutan yang sama.

Maka nilai Jaro distance adalah:

$$d_j = \frac{1}{3} \times \left(\frac{4}{6} + \frac{6}{5} + \frac{4-1}{4} \right) = 0.822$$

Kemudian bila diperhatikan susunans_1 dan s_2 dapat diketahui nilai l *=1*, dan dengan nilai konstan p*= 0.1*. Maka nilai Jaro-Winkler distance adalah:

$$d_w = 0.822 + (1 \times 0.1\,(1 - 0.8.22)) = 0.961$$

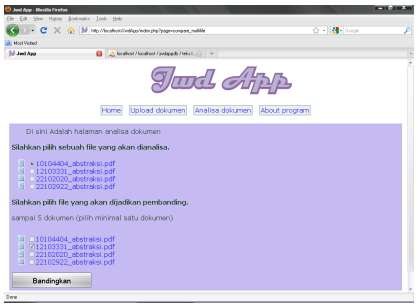
3 Metedologi

Data yang digunakan adalah data abstraksi dari penulisan ilmiah mahasiswa jurusan Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma. Data yang digunakan berjumlah 50 dokumen yang bertipe pdf. Data diambil dari perpustakaan online Universitas Gunadarma yang beralamat di http://library.gunadarma.ac.id. Akan tetapi, dokumen yang menjadi bahan untuk pengujian pada aplikasi adalah dokumen-dokumen yang telah dipilih untuk mewakili berbagai tingkat kesamaan.

4 Hasil dan Pembahasan

4.1 Tampilan Program

Aplikasi yang dikembangkan untuk menghitung tingkat kesamaan dokumen ini, memiliki menu Home, Upload Dokumen, Analisa Dokumen dan About Program. Menu Upload Dokumen digunakan untuk memasukkan atau menambahkan dokumen yang akan di bandingkan. Menu Analisa Dokumen digunakan untuk menganalisa atau membandingkan dokumen. Tampilan output dari aplikasi yang dikembangkan adalah seperti pada gambar 1 berikut ini.

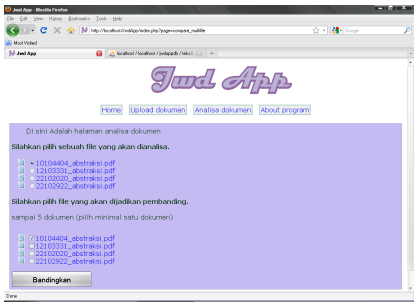


Gambar 1: Tampilan Output Aplikasi

Aplikasi diuji dengan beberapa dokumen dengan berbagai tingkat kesamaan, baik melalui dokumen yang telah dipilih acak maupun tidak. Untuk mengetahui ketepatan pemeriksaan salinan dipergunakan dokumen yang sama, sedangkan pada dua uji coba lainnya akan dipilih file yang memiki kesamaan tinggi tapi berbeda urutan dan juga dipilih dokumen banyak memiliki kata dengan urutan yang sama.

4.1.1 Pengujian Dengan Dokumen Yang Sama

Pada pengujian ini aplikasi akan menggunakan dokumen yang sama sebagai dokumen yang dianalisis maupun dokumen pembanding. Diharapkan hasil analisis akan mendapatkan kecocokan seratus persen, yang berarti dokumen yang dianalisa identik/sama persis dengan dokumen pembanding. Gambar 2 akan memperlihatkan dokumen yang dipilih. Gambar 3akan memperlihatkan hasil dokumen.

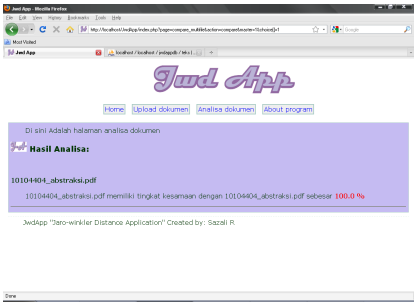


Gambar 2: Pengujian dokumen sama

4.2 Hasil Uji Coba dan Analisis

Hasil dari pengujian dari kinerja aplikasi dapat dilihat pada tabel 1 berikut:

Hasil analisis :



Gambar 3: Hasil analisa Dokumen

Tabel 1: Hasil uji coba aplikasi

Identik	File yang dibandingkan	Keterangan
Identik	10104404_abstraksi.pdf dan10104404_abstraksi.pdf	Berhasil
Mirip isi tektual	22102922_abstraksi.pdf dan 22102020_abstraksi.pdf	Gagal
Mirip dengan urutan kata yang sama	10104039_abstraksi.pdf dan 11104401_abstraksi.pdf	Berhasil

1. 1.Pada dokumen yang sama hasilnya memuaskan, hal ini terjadi dikarenakan data yang dibandingkan memiliki urutan yang sama persis dan juga isi tekstualnya sama persis.
2. Pada ujicoba kedua yang menyebabkan kegagalan adalah berbedanya urutan kata sehingga walau isi tekstualnya sama, akan tetapi aplikasi tidak dapat mendeteksinya.
3. Pada ujicoba ketiga dikarenakan memiliki banyak kata memiliki urutan yang sama, sehingga walau dilihat secara tekstual berbeda, aplikasi masih dapat mendeteksi kesamaan.

5 Kesimpulan

Aplikasi yang dibuat telah berhasil menggunakan algoritma Jaro-Winkler distance untuk mendukung kinerjanya. Dalam ujicobanya aplikasi dapat berjalan dengan baik untuk memeriksa kemiripan dokumen yang identik atau sama seratus persen. Hal ini dikarenakan urutan kata-kata yang dibandingkan sangat sesuai. Akan tetapi, saat memeriksa kemiripan dokumen dengan urutan yang berbeda, aplikasi ini tidak mampu mendeteksi kemiripannya. Hal ini juga dikarenakan urutan kata yang telah berbeda pula.

Maka bisa disimpulkan bahwa aplikasi ini berjalan baik untuk dokumen yang memiliki kemiripan dan urutan kata yang sama.

Pustaka

[1] I Wayan Simri W Ana Kurniawati. Perbandingan pendekatan deteksi plagiarism dokumen dalam bahasa inggris. In *Proceeding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008)*, Agustus 2008.

[2] David Sugianto. *membangun Websited dengan PHP*. Datakom, 2005.

[3] Felicia Utorodewo. *Bahasa Indonesia : Sebuah Pengantar Penulisan Ilmiah*. Penerbit Fakultas Ekonomi Universitas Indonesia, 2007.

[4] W. E. Winkler. The state of record linkage and current research problems. *Statistics of Income Division, Internal Revenue Service Publication R99/04.*, 1999.

[5] W. E. Winkler. Overview of record linkage and current research directions. *Research Report Series, RRS*, 2006.